

WHAT IS CLAIMED

1. A method of snapshot management in a data storage system that maintains multiple snapshot versions and an active file system, comprising:
 - (a) maintaining an index table that contains an entry for each snapshot and the active file system; and
 - (b) maintaining a space map block entry (b, e), wherein b and e represent index table entries, b indicates a first snapshot that uses the first block, and e indicates a last snapshot that uses the first block.
2. The method of claim 1, further comprising:
 - (c) allocating a second block;
 - (d) writing the data in the second block;
 - (e) updating any other blocks that pointed to the first block to point to the second block; and
 - (f) repeating the steps (c) through (e) for updating other blocks.
3. The method of claim 1, wherein the index table entry includes one or more attributes selected from a group of attributes comprising a version number, an image state, a time stamp, a root block pointer, and an image name.
4. The method of claim 2, further comprising (g) reading the associated space map block to determine if the first block is only used for the active file system and if so, updating the first block.
5. The method of claim 1, wherein the space map block entry (b, e) are indexes into the index table.
6. The method of claim 1, wherein the space map block entry (b, e) are version numbers in the index table.

7. The method of claim 4, wherein the space map block entry (a, 0) indicates the first block is only used by the active file system, wherein "a" represents the active file system.

8. The method of claim 1, further comprising:

- (c) reading a first index table entry associated with the active file system;
- (d) locating a second index table entry that is free-to-use;
- (e) assigning a version number to the second index table entry that is larger than that of the first index table entry;
- (f) copying the root block pointer from the first index table to the second index table entry;
- (g) setting the image state of the first index table entry to an in-use snapshot; and
- (h) setting the image state of the second index table entry to the active file system to create a snapshot.

9. The method of claim 1, further comprising:

- (c) maintaining in the index table an image state that indicates if an entry is deleted or not; and
- (d) marking the image state as deleted to delete a snapshot.

10. A method of claim 1, further comprising:

- (c) ordering the index table by time of snapshot creation;
- (d) reading each space map block entry to determine if the first snapshot was deleted and if so, replacing the index b with the earliest index of a later undeleted snapshot or the active file system, determining if the last snapshot was deleted and if so, replacing the index e with the latest index of an earlier undeleted snapshot to clean deleted snapshots.

11. The method of claim 10, further comprising (e) resetting the image state to free for all index table entries of deleted snapshots when all space map block entries have been processed.

12. The method of claim 10, further comprising (e) setting the space map block entry to indicate that the associated block is free if no undeleted snapshot exists between index b and index e.

13. The method of claim 12, further comprising (f) resetting the image state to free for all index table entries of deleted snapshots when all space map block entries have been processed.

14. The method of claim 1, further comprising:
(c) maintaining in the index table an image state that indicates if an entry is obsolete or not;
(d) creating a new snapshot c;
(e) creating an active file system a; and
(f) setting the image state of all in-use snapshots before snapshot c and after snapshot p to obsolete to revert to a previous snapshot p.

15. The method of claim 1, further comprising:
(c) maintaining in the index table an image state that indicates if an entry is obsolete or not;
(d) reading a snapshot c;
(e) reading a snapshot p;
(f) reading each space map block entry to determine:
(i) if the first snapshot is snapshot p or earlier than snapshot p and the last snapshot is earlier than snapshot c, replacing the index e with a value indicating that the block is still in use by the active file system,

(ii) if the first snapshot is later than snapshot p and earlier than snapshot c, setting the space map block entry to indicate that the associated block is free;

(g) marking the image state of snapshot c as deleted; and

(h) resetting the image state of all obsolete index table entries to free to clean obsolete snapshots after a revert operation to snapshot p.

16. The method of claim 15, further comprising repeating steps (f) and (g) over a list of snapshots p and of snapshots c

17. The method of claim 16, wherein the pending list includes the reverted snapshots p and the created snapshots c.

18. A snapshot management system including a processor for maintaining multiple snapshot versions and an active file system, comprising:

an index table that contains an entry for each snapshot and the active file system;

a space map block entry (b, e), wherein b and e represent index table entries, b indicates a first snapshot that uses the first block, and e indicates a last snapshot that uses the first block; and

a usable space for storing the snapshot versions and the active file system.

19. The system of claim 18, wherein the processor is programmed to execute steps of:

allocating a second block;

writing the data in the second block;

updating any other blocks that pointed to the first block to point to the second block; and

repeating the above steps for updating other blocks.

20. The system of claim 18, wherein the index table entry includes one or more attributes selected from a group of attributes comprising a version number, an image state, a time stamp, a root block pointer, and an image name.

21. The system of claim 19, wherein the processor reads the associated space map block to determine if the first block is only used for the active file system and if so, updates the first block.

22. The system of claim 18, wherein the space map block entry (b, e) are indexes into the index table.

23. The system of claim 18, wherein the space map block entry (b, e) are version numbers in the index table.

24. The system of claim 18, wherein the space map block entry (a, 0) indicates the first block is only used by the active file system, wherein "a" represents the active file system.

25. The system of claim 18, wherein the processor is programmed to execute steps of:

- reading a first index table entry associated with the active file system;
- locating a second index table entry that is free-to-use;
- assigning a version number to the second index table entry that is larger than that of the first index table entry;
- copying the root block pointer from the first index table to the second index table entry;
- setting the image state of the first index table entry to an in-use snapshot;
- and
- setting the image state of the second index table entry to the active file system to create a snapshot.

26. The system of claim 18, wherein the processor is programmed to execute steps:

maintaining in the index table an image state that indicates if an entry is deleted or not; and

marking the image state as deleted to delete a snapshot.

27. The system of claim 18, wherein the processor is programmed to execute steps:

ordering the index table by time of snapshot creation;

reading each space map block entry to determine if the first snapshot was deleted and if so, replacing the index b with the earliest index of a later undeleted snapshot or the active file system, determining if the last snapshot was deleted and if so, replacing the index e with the latest index of an earlier undeleted snapshot to clean deleted snapshots.

28. The system of claim 27, wherein the processor is programmed to reset the image state to free for all index table entries of deleted snapshots when all space map block entries have been processed.

29. The system of claim 27, wherein the processor is programmed to set the space map block entry to indicate the associated block is free if no undeleted snapshot exists earlier than index e or if index b is later than index e.

30. The system of claim 29, wherein the processor is programmed to reset the image state to free for all index table entries of deleted snapshots when all space map block entries have been processed.

31. The system of claim 18, wherein the processor is programmed to execute:

maintaining in the index table an image state that indicates if an entry is obsolete or not;

- creating a new snapshot c;
- creating an active file system a; and
- setting the image state of all in-use snapshots before snapshot c and after snapshot p to obsolete to revert to a previous snapshot p.

32. The system of claim 27, wherein the processor is programmed to set the space map block entry to indicate that the associated block is free if no undeleted snapshot exists earlier than index e or if index b is later than index e.

33. The system of claim 18, wherein the processor is programmed to execute:

- maintaining in the index table an image state that indicates if an entry is obsolete or not;

- reading a snapshot c;

- reading a snapshot p;

- reading each space map block entry to determine:

- (i) if the first snapshot is snapshot p or earlier than snapshot p and the last snapshot is earlier than snapshot c, replacing the index e with a value indicates that the block is still in use by the active file system,

- (ii) if the first snapshot is later than snapshot p and earlier than snapshot c, setting the space map block entry to indicate that the associated block is free;

- marking the image state of snapshot c as deleted; and

- resetting the image state of all obsolete index table entries to free to clean obsolete snapshots after a revert operation to snapshot p.

34. The system of claim 33, wherein the processor is programmed to repeat the reading and marking over a list of snapshots p and of snapshots c

35. The system of claim 34, wherein the pending list includes the reverted snapshots p and the created snapshots c.

36. A snapshot management system that maintains multiple snapshot versions and an active file system, comprising:

means for maintaining an index table that contains an entry for each snapshot and the active file system; and

means for maintaining space map block entry (b, e), wherein b and e represent index table entries, b indicates a first snapshot that uses the first block, and e indicates a last snapshot that uses the first block.

37. The system of claim 36, further comprising:

means for allocating a second block;

means for writing the data in the second block;

means for updating any other blocks that pointed to the first block to point to the second block; and

means for repeating the above steps for updating other blocks.

38. The system of claim 36, wherein the index table entry includes one or more attributes selected from a group of attributes comprising a version number, an image state, a time stamp, a root block pointer, and an image name.

39. The system of claim 37, further comprising means for reading the associated space map block to determine if the first block is only used for the active file system and if so, updating the first block.

40. The system of claim 36, wherein the space map block entry (b, e) are indexes into the index table.

41. The system of claim 36, wherein the space map block entry (b, e) are version numbers in the index table.

42. The system of claim 39, wherein the space map block entry (a, 0) indicates the first block is only used by the active file system, wherein "a" represents the active file system.

43. The system of claim 36, further comprising:
means for reading a first index table entry associated with the active file system;
means for locating a second index table entry that is free-to-use;
means for assigning a version number to the second index table entry that is larger than that of the first index table entry;
means for copying the root block pointer from the first index table to the second index table entry;
means for setting the image state of the first index table entry to an in-use snapshot; and
means for setting the image state of the second index table entry to the active file system to create a snapshot.

44. The system of claim 36, further comprising:
means for maintaining in the index table an image state that indicates if an entry is deleted or not; and
means for marking the image state as deleted to delete a snapshot.

45. The system of claim 36, further comprising:
means for ordering the index table by time of snapshot creation; and
means for reading each space map block entry to determine if the first snapshot was deleted and if so, replacing the index b with the earliest index of a later undeleted snapshot or the active file system, determining if the last snapshot was deleted and if so, replacing the index e with the latest index of an earlier undeleted snapshot to clean deleted snapshots.

46. The system of claim 45, further comprising means for resetting the image state to free of all index table entries of deleted snapshots when all space map block entries have been processed.

47. The system of claim 45, further comprising means for setting the space map block entry to indicate that the associated block is free if no undeleted snapshot exists earlier than index e or if index b is later than index e.

48. The system of claim 47, further comprising means for resetting the image state to free of all index table entries of deleted snapshots when all space map block entries have been processed.

49. The system of claim 36, further comprising:
means for maintaining in the index table an image state that indicates if an entry is obsolete or not;
means for creating a new snapshot c;
means for creating an active file system a; and
means for setting the image state of all in-use snapshots before snapshot c and after snapshot p to obsolete to revert to a previous snapshot p.

50. The system of claim 36, further comprising:
means for maintaining in the index table an image state that indicates if an entry is obsolete or not;
means for reading a snapshot c;
means for reading a snapshot p;
means for reading each space map block entry to determine:
(i) if the first snapshot is snapshot p or earlier than snapshot p and the last snapshot is earlier than snapshot c, replacing the index e with a value indicating that the block is still in use by the active file system,

(ii) if the first snapshot is later than snapshot p and earlier than snapshot c, setting the space map block entry to indicate that the associated block is free;

means for marking the image state of snapshot c as deleted; and

means for resetting the image state of all obsolete index table entries to free to clean obsolete snapshots after a revert operation to snapshot p.

51. The system of claim 50, further comprising means for repeating the reading and marking functions over a list of snapshots p and of snapshots c

52. The system of claim 51, wherein the list includes the unclean reverted snapshots p and the created snapshots c.